```
0000:          * EXPLORER CONTROL PROGRAM
0000:          * COPYRIGHT 1982 BY FROBCO -- ALL RIGHTS RESERVED
0000:          *
0000:          * VERSION  1.3 -- LAST MODIFIED 2/2/83
0000:          *
0000:          *
0000:          * SYSTEM DEFINITIONS
C000:          KBD     EQU  $C000     ; APPLE KEYBOARD REGISTER
C010:          KSTRB   EQU  $C010     ; APPLE KEYBOARD STROBE LOCATION
0010:          SACMD   EQU  $10       ; SET ADDRESS COMMAND FOR FROB
0020:          RDCMD   EQU  $20       ; READ REGISTER COMMAND
0040:          WRCMD   EQU  $40       ; WRITE REGISTER COMMAND
0000:          *
0000:          *
0000:          *
0000:          * PAGE 0 RAM ALLOCATION
0000:          *
0085:          RLOC    EQU  $85       ; READ REGISTER ADDRESS
0086:          FSTAT   EQU  RLOC+1    ; POINTER TO FROB STATUS REGISTER
0088:          FDATA   EQU  FSTAT+2   ; POINTER TO FROB DATA REGISTER
008A:          RBASE   EQU  FDATA+2   ; SCREEN ROW BASE POINTER FOR READ REGISTERS
008C:          CRBASE  EQU  RBASE+2   ; SCREEN ROW BASE POINTER USED BY CURSOR
008E:          FBASE   EQU  CRBASE+2  ; FIELD ARRAY ROW BASE POINTER
0090:          CROW    EQU  FBASE+2   ; CURSOR ROW
0091:          CCOL    EQU  CROW+1    ; CURSOR COLUMN
0092:          RRCOL   EQU  CCOL+1    ; READ REG COLUMN
0093:          OLDCR   EQU  RRCOL+1   ; OLD CURSOR ROW
0094:          OLDCCL  EQU  OLDCR+1   ; OLD CURSOR COLUMN
0095:          ROW     EQU  OLDCCL+1  ; PARAMETER TO ROW BASE LOOKUP ROUTINE
0096:          TMP1    EQU  ROW+1     ; RANDOM TEMP STORE
0097:          RDHGH   EQU  TMP1+1    ; READ ADDRESS HIGH POINTER
0098:          RDLOW   EQU  RDHGH+1   ; LOW PART OF ABOVE
0099:          LSTKEY  EQU  RDLOW+1   ; LAST KEY PRESSED
009A:          LSTRD   EQU  LSTKEY+1  ; LAST READ REGISTER VALUE
009B:          MASK    EQU  LSTRD+1   ; MASK TO USE ON THE READBACK LOCATIONS
0000:          *
0000:          *
0000:          * START OF PROGRAM -- LOAD AT $8000
0000:          *
----- NEXT OBJECT FILE NAME IS XCONTROL.OBJ0
8000:                  ORG  $8000
8000:          *
8000:20 15 83  START   JSR  SAVER     ; THIS PROGRAM IS CALLED FROM BASIC
8003:20 B7 80  MLOOP   JSR  INCRR     ; GO INCREMENT THE READ REGISTER
8006:90 03             BCC  MLP1      ; CHECK FOR FULL SET DONE
8008:20 74 80          JSR  SPECR     ; IF SO GO DO SPECIAL READ
800B:20 E7 80  MLP1    JSR  RDREG     ; GO DO STANDARD READ
800E:85 9A             STA  LSTRD     ; SAVE RESULT
8010:20 FB 80          JSR  ADRCNV    ; DO AN ADDRESS CONVERSION SO WE KNOW WHERE THIS REGISTER
8013:          *        SHOULD GO ON THE SCREEN.
8013:A5 9A             LDA  LSTRD     ; GET VALUE BACK
8015:25 9B             AND  MASK      ; MASK WITH USER SELECTED MASK
8017:20 34 81          JSR  DSPBYT    ; PUT IT UP
801A:20 58 81          JSR  CHKKEY    ; CHECK FOR KEY PRESSED (GET IT IF SO)
801D:90 E4             BCC  MLOOP     ; END OF MAIN LOOP IF NO KEY
801F:C9 83             CMP  #$83      ; SEE IF CONTROL C
8021:D0 03             BNE  MLP2      ; IF SO EXIT TO BASIC
8023:4C 22 83          JMP  RESTORER
8026:85 99     MLP2    STA  LSTKEY    ; SAVE KEY
8028:C9 BB             CMP  #$BB      ; IS IT THE ";" (UP)?
802A:D0 06             BNE  NTUP      ; IF NOT GO ON
802C:20 67 81          JSR  MOVUP     ; ELSE MOVE CURSOR UP
802F:4C 03 80          JMP  MLOOP
8032:C9 AF     NTUP    CMP  #$AF      ; IS IT THE "/" (DOWN)?
8034:D0 06             BNE  NTDWN     ; IF NOT GO ON
8036:20 7C 81          JSR  MOVDWN    ; ELSE MOVE CURSOR DOWN
8039:4C 03 80          JMP  MLOOP
803C:C9 88     NTDWN   CMP  #$88      ; IS IT THE LEFT ARROW?
803E:D0 06             BNE  NTLFT     ; IF NOT GO ON
8040:20 A8 81          JSR  MOVLFT    ; ELSE MOVE CURSOR LEFT
8043:4C 03 80          JMP  MLOOP
8046:C9 95     NTLFT   CMP  #$95      ; IS IT THE RIGHT ARROW?
8048:D0 06             BNE  NTRGH     ; IF NOT GO ON
804A:20 90 81          JSR  MOVRGH    ; ELSE MOVE CURSOR RIGHT
804D:4C 03 80          JMP  MLOOP
8050:          *
```

```
8050:20 EF 81   NTRGH   JSR  NUMFLD    ; CHECK TO SEE IF CURSOR IN NUMBER FIELD
8053:90 AE              BCC  MLOOP     ; IF NOT THE KEY IS MEANINGLESS
8055:A5 99              LDA  LSTKEY    ; GET KEY BACK
8057:20 1F 81           JSR  HEXKEY    ; IS IT A HEXIDECIMAL DIGIT?
805A:90 A7              BCC  MLOOP     ; IF NOT THE KEY IS MEANINGLESS
805C:A5 99              LDA  LSTKEY    ; GET KEY BACK AGAIN
805E:20 2C 82           JSR  DSPCHR    ; GO PUT IT AT CURSOR AND ADVANCE CURSOR
8061:20 90 81           JSR  MOVRGH
8064:20 EF 81           JSR  NUMFLD    ; SEE IF STILL IN NUMBER FIELD
8067:B0 9A              BCS  MLOOP     ; IF SO WE ARE DONE FOR NOW
8069:20 04 82           JSR  SPECCK    ; ELSE TIME TO DO SOMETHING SO CHECK FOR SPECIAL CASE
806C:B0 95              BCS  MLOOP     ; IF TRUE THEN THE SPECIAL CASE HAS BEEN DONE BEFORE RETURN
806E:20 3A 82           JSR  DOWRT     ; GO DO THE VCS REGISTER WRITE
8071:4C 03 80           JMP  MLOOP     ; END OF TOP LEVEL
8074:           *
8074:           *
8074:           * ROUTINE TO DO A SPECIAL READ
8074:           *
8074:A5 93      SPECR   LDA  OLDCR     ; WE WILL NEED TO USE
8076:48                 PHA             ; OLDCR AND OLDCCL AS PARAMETERS
8077:A5 94              LDA  OLDCCL    ; TO GETVAL
8079:48                 PHA
807A:A9 12              LDA  #$12      ; ROW OF SPECIAL OPERATIONS
807C:85 93              STA  OLDCR
807E:A9 13              LDA  #$13      ; COLUMN OF SECOND DIGIT OF THE HIGH ORDER ADDRESS
8080:85 94              STA  OLDCCL
8082:20 5E 82           JSR  GETVAL    ; GET HIGH ADDRESS PART
8085:85 97              STA  RDHGH     ; PUT IN HIGH ADDR POINTER
8087:A9 15              LDA  #$15      ; COLUMN OF SECOND DIGIT OF THE LOW ORDER ADDRESS
8089:85 94              STA  OLDCCL    ; PASS ON TO GETVAL
808B:20 5E 82           JSR  GETVAL
808E:85 98              STA  RDLOW     ; NOW THE ADDRESS IS SET UP
8090:68                 PLA             ; GET OLDCCL BACK
8091:85 94              STA  OLDCCL
8093:68                 PLA             ; GET OLDCR BACK
8094:85 93              STA  OLDCR
8096:20 E7 80           JSR  RDREG     ; GO DO THE VCS READ
8099:48                 PHA             ; SAVE THE RESULT
809A:A9 24              LDA  #$24      ; COLUMN OF THE RESULT
809C:85 92              STA  RRCOL
809E:A2 12              LDX  #$12      ; ROW OF THE RESULT
80A0:BD 2F 83           LDA  CHLIST,X  ; GET HIGH PART OF SCREE ROW
80A3:85 8B              STA  RBASE+1   ; PUT IN HIGH PART
80A5:BD 47 83           LDA  CLLIST,X  ; GET LOW PART
80A8:85 8A              STA  RBASE
80AA:68                 PLA             ; GET VALUE TO BE DISPLAYED
80AB:20 34 81           JSR  DSPBYT    PUT IT UP
80AE:A9 00              LDA  #0        ; CLEAR RDHGH
80B0:85 97              STA  RDHGH
80B2:A5 85              LDA  RLOC      ; RESTORE RDLOW
80B4:85 98              STA  RDLOW
80B6:60                 RTS
80B7:           *
80B7:           *
80B7:           * ROUTINE TO INCREMENT READ REGISTER NUMBER AND RETURN CARRY SET
80B7:           * WHEN THROUGH THE ENTIRE LIST
80B7:           *
80B7:E6 85      INCRR   INC  RLOC
80B9:A5 85              LDA  RLOC      ; GET THE NEW VALUE
80BB:C9 40              CMP  #$40      ; GONE TOO FAR?
80BD:90 04              BCC  INCR1     ; BRANCH IF NOT AT END
80BF:A9 30              LDA  #$30      ; ELSE RESET COUNTER
80C1:85 85              STA  RLOC
80C3:A9 00      INCR1   LDA  #0        ; UPDATE THE READ ADDRESS POINTERS
80C5:85 97              STA  RDHGH     ; HIGH PART
80C7:A5 85              LDA  RLOC
80C9:85 98              STA  RDLOW     ; LOW PART
80CB:60                 RTS                  ; DONE
80CC:           *
80CC:           *
80CC:           * ROUTINE TO DO A SET READ/WRITE ADDRESS OVER IN THE VCS
80CC:           *
80CC:A0 00      SETADR  LDY  #0        ; SO WE CAN LOAD INDIRECT
80CE:B1 86      SA1     LDA  (FSTAT),Y ; GO GET FROB STATUS REGISTER
80D0:10 FC              BPL  SA1       ; WAIT FOR OK TO WRITE
80D2:A9 10              LDA  #SACMD    ; GET THE SET ADDRESS COMMAND CODE
80D4:91 88              STA  (FDATA),Y ; SEND IT TO THE FROB EXPLORER
```

```
80D6:B1 86     SA2     LDA  (FSTAT),Y ; GO GET FROB STATUS REGISTER
80D8:10 FC             BPL  SA2       ; WAIT FOR OK FOR NEXT WRITE
80DA:A5 97             LDA  RDHGH     ; GET THE HIGH ORDER PART
80DC:91 88             STA  (FDATA),Y ; SEND IT
80DE:B1 86     SA3     LDA  (FSTAT),Y ; SAME AS ABOVE
80E0:10 FC             BPL  SA3
80E2:A5 98             LDA  RDLOW     ; GET THE LOW ORDER PART
80E4:91 88             STA  (FDATA),Y ; THAT FINISHES THE JOB.
80E6:60                RTS
80E7:          *
80E7:          *
80E7:          * ROUTINE TO READ A VCS REGISTER
80E7:          *
80E7:20 CC 80  RDREG   JSR  SETADR    ; SEND THE VCS THE ADDRESS OF THE REGISTER WE WANT
80EA:B1 86     RDR0    LDA  (FSTAT),Y ; GET FROB STATUS
80EC:10 FC             BPL  RDR0      ; WAIT TIL OK TO WRITE
80EE:A9 20             LDA  #RDCMD    ; SEND IT A READ REGISTER COMMAND
80F0:91 88             STA  (FDATA),Y
80F2:B1 86     RDR1    LDA  (FSTAT),Y ; GET FROB STATUS REGISTER
80F4:29 40             AND  #$40      ; LOOK AT DATA COMING BACK BIT
80F6:F0 FA             BEQ  RDR1      ; WAIT FOR DATA TO COME BACK
80F8:B1 88             LDA  (FDATA),Y ; NOW READ DATA
80FA:60                RTS
80FB:          *
80FB:          *
80FB:          * ROUTINE TO CONVERT READ REGISTER NUMBER INTO SCREEN ADDRESS
80FB:          *
80FB:A5 85     ADRCNV  LDA  RLOC      ; GET READ REGISTER NUMBER
80FD:29 0F             AND  #$0F      ; LOOK AT LOWER BITS
80FF:4A                LSR  A          ; DIVIDE BY 4
8100:4A                LSR  A
8101:09 14             ORA  #$14      ; PUT IN ROW OFFSET
8103:AA                TAX            ; SAVE IN X
8104:BD 2F 83          LDA  CHLIST,X  ; LOOKUP THE ROW BASE ADDRESS HIGH
8107:85 8B             STA  RBASE+1   ; SAVE IT IN POINTER
8109:BD 47 83          LDA  CLLIST,X  ; LOOKUP THE ROW BASE ADRESS LOW
810C:85 8A             STA  RBASE     ; SAVE IT IN POINTER
810E:A5 85             LDA  RLOC      ; GET REGISTER NUMBER AGAIN
8110:29 03             AND  #$3       ; LOOK AT LAST TWO BITS
8112:85 96             STA  TMP1      ; MULTIPLY BY 9
8114:0A                ASL  A
8115:0A                ASL  A
8116:0A                ASL  A
8117:05 96             ORA  TMP1
8119:18                CLC                 ; NOW ADD COLUMN OFFSET
811A:69 06             ADC  #6
811C:85 92             STA  RRCOL
811E:60                RTS
811F:          *
811F:          *
811F:          * ROUTINE TO CHECK KEY NUMBER FOR LEGAL HEX DIGIT
811F:          *
811F:C9 BA     HEXKEY  CMP  #$BA      ; SEE IF IT IS IN ALPHA RANGE
8121:B0 05             BCS  HK1
8123:C9 B0             CMP  #$B0
8125:90 0B             BCC  NOHEX     ; ALSO FAIL IF TOO LOW
8127:60                RTS                 ; ELSE OK
8128:          *
8128:C9 C1     HK1     CMP  #$C1      ; SEE IF OUT OF BA>C0 RANGE
812A:90 06             BCC  NOHEX
812C:C9 C7             CMP  #$C7      ; SEE IF OFF THE TOP
812E:B0 02             BCS  NOHEX
8130:38                SEC               ; ELSE IT IS OK
8131:60                RTS
8132:          *
8132:18        NOHEX   CLC            ; RETURN CARRY CLEAR ON FAIL
8133:60                RTS
8134:          *
8134:          *
8134:          * ROUTINE TO PUT UP HEX BYTE AT RBASE AND RRCOL
8134:          *
8134:85 96     DSPBYT  STA  TMP1      ; SAVE BYTE TO BE DISPALYED
8136:4A                LSR  A          ; GET UPPER 4 BITS
8137:4A                LSR  A
8138:4A                LSR  A
8139:4A                LSR  A
813A:20 4C 81          JSR  TOHEX     ; CONVERT NIBBLE TO ALPHANUMERIC FOR THE HEX DIGIT
```

```
813D:A4 92             LDY  RRCOL     ; GET THE COLUMN ADDRESS
813F:91 8A             STA  (RBASE),Y ; PUT THE CHAR ON THE SCREEN
8141:A5 96             LDA  TMP1      ; GET BYTE BACK
8143:20 4C 81          JSR  TOHEX     ; CONVERT LOWER NIBBLE
8146:A4 92             LDY  RRCOL     ; GET THE COLUMN ADDRESS
8148:C8                INY             ; MOVE TO NEXT COLUMN
8149:91 8A             STA  (RBASE),Y ; PUT UP CHAR
814B:60                RTS
814C:          *
814C:          *
814C:          * ROUTINE TO RETURN HEX ALPHA CHAR FOR LOW NIBBLE IN A
814C:          *
814C:29 0F     TOHEX   AND  #$0F      ; GET LOW NIBBLE
814E:C9 0A             CMP  #$0A      ; SEE IF IN THE ALPHA GROUP
8150:90 02             BCC  TH1       ; BRANCH IF NOT ALPHA
8152:69 06             ADC  #6        ; IF SO ADD 7 (CARRY + 6)
8154:18        TH1     CLC            ; PUT IN UPPER BITS
8155:69 B0             ADC  #$B0
8157:60                RTS
8158:          *
8158:          * ROUTINE TO SEE IF KEY PRESSED (RETURNS VALUE AND CARRY SET IF SO)
8158:          *
8158:2C 00 C0  CHKKEY  BIT  KBD       ; LOOK AT KBD STROBE BIT
815B:30 02             BMI  CKY1      ; BRANCH IF KEY PRESSED
815D:18                CLC              ; ELSE CLEAR CARRY AND EXIT
815E:60                RTS
815F:          *
815F:AD 00 C0  CKY1    LDA  KBD       ; GET KEY CODE
8162:2C 10 C0          BIT  KSTRB     ; RESET THE STROBE BIT
8165:38                SEC              ; SHOW WE GOT IT
8166:60                RTS
8167:          *
8167:          *
8167:          * ROUTINE TO MOVE THE CURSOR UP
8167:          *
8167:A5 90     MOVUP   LDA  CROW      ; GET THE CURSOR ROW POSITION
8169:85 93             STA  OLDCR     ; THIS IS NOW THE OLD POSITION
816B:C6 90             DEC  CROW      ; MOVE NEW POSITION UP
816D:A5 90             LDA  CROW      ; CHECK FOR WRAP OFF TOP\021
816F:10 04             BPL  MUP1      ; BRANCH IF OK
8171:A9 17             LDA  #$17      ; ELSE MOVE TO BOTTOM ROW
8173:85 90             STA  CROW
8175:A5 91     MUP1    LDA  CCOL      ; UPDATE OLD COLUMN
8177:85 94             STA  OLDCCL
8179:4C BA 81          JMP  CUPDTE    ; THEN GO UPDATE THE CURSOR
817C:          *
817C:          *
817C:          * ROUTINE TO MOVE CURSOR DOWN
817C:          *
817C:A5 90     MOVDWN  LDA  CROW      ; GET THE CURSOR ROW
817E:85 93             STA  OLDCR     ; UPDATE THE OLD POSITION
8180:C9 17             CMP  #$17      ; AT THE BOTTOM?
8182:F0 05             BEQ  MDN1      ; IF SO WRAP TO TOP
8184:E6 90             INC  CROW      ; ELSE MOVE DOWN
8186:4C 75 81          JMP  MUP1      ; AND HANDLE AS ABOVE
8189:          *
8189:A9 00     MDN1    LDA  #0        ; WRAP TO TOP OF SCREEN
818B:85 90             STA  CROW
818D:4C 75 81          JMP  MUP1
8190:          *
8190:          *
8190:          * ROUTINE TO MOVE THE CURSOR TO THE RIGHT
8190:          *
8190:A5 91     MOVRGH  LDA  CCOL      ; GET THE CURSOR COLUMN
8192:85 94             STA  OLDCCL    ; UPDATE THE OLD POSITION
8194:C9 27             CMP  #39       ; SEE IF AT RIGHT EDGE
8196:D0 07             BNE  MRT0      ; BRANCH IF SIMPLE MOVE
8198:A9 00             LDA  #0        ; ELSE RESET COLUMN
819A:85 91             STA  CCOL
819C:4C A1 81          JMP  MRT1
819F:          *
819F:E6 91     MRT0    INC  CCOL      ; ELSE MOVE TO THE RIGHT
81A1:A5 90     MRT1    LDA  CROW      ; GET THE ROW
81A3:85 93             STA  OLDCR      ; UPDATE OLD POSITION
81A5:4C BA 81          JMP  CUPDTE    ; GO UPDATE THE CURSOR
81A8:          *
81A8:          *
```

```
81A8:           * ROUTINE TO MOVE THE CURSOR LEFT
81A8:           *
81A8:A5 91      MOVLFT  LDA  CCOL       ; GET THE COLUMN
81AA:85 94              STA  OLDCCL     ; UPDATE OLD POSITION
81AC:D0 07              BNE  MLF1       ; BRANCH IF ROOM FOR SIMPLE MOVE
81AE:A9 27              LDA  #39        ; ELSE MOVE TO LAST COLUMN
81B0:85 91              STA  CCOL
81B2:4C A1 81           JMP  MRT1
81B5:           *
81B5:C6 91      MLF1    DEC  CCOL
81B7:4C A1 81           JMP  MRT1       ; TAKE CARE OF AS ABOVE
81BA:           *
81BA:           *
81BA:           * ROUTINE TO UPDATE CURSOR POSITION
81BA:           *
81BA:A4 94      CUPDTE  LDY  OLDCCL     ; GET OLD CURSOR COLUMN POSITION
81BC:A5 93              LDA  OLDCR      ; GET OLD CURSOR ROW
81BE:85 95              STA  ROW        ; SAVE AS PARAMETER TO CADDR
81C0:20 E2 81           JSR  CRADDR     ; LOOKUP CURSOR ROW BASE ADDRESS GIVEN "ROW"
81C3:B1 8C              LDA  (CRBASE),Y ; GET WHAT IS AT THE OLD CURSOR POSITION
81C5:09 80              ORA  #$80       ; TAKE OUT OF INVERSE VIDEO
81C7:85 96              STA  TMP1       ; HERE IS THE MAPPING
81C9:29 20              AND  #$20       ; ALGORITHM
81CB:0A                 ASL  A
81CC:49 40              EOR  #$40
81CE:05 96              ORA  TMP1
81D0:91 8C              STA  (CRBASE),Y ; PUT IT BACK
81D2:A4 91              LDY  CCOL       ; GET THE NEW CURSOR COLUMN
81D4:A5 90              LDA  CROW       ; GET THE NEW CURSOR ROW
81D6:85 95              STA  ROW        ; SAVE AS PARAMETER TO CRADDR
81D8:20 E2 81           JSR  CRADDR     ; LOOKUP NEW CURSOR ROW BASE ADDRESS GIVEN "ROW"
81DB:B1 8C              LDA  (CRBASE),Y ; GET NEW CURSOR CHARACTOR
81DD:29 3F              AND  #$3F       ; MAKE INVERSE VIDEO
81DF:91 8C              STA  (CRBASE),Y ; PUT IT BACK
81E1:60                 RTS
81E2:           *
81E2:           *
81E2:           * ROUTINE TO LOOKUP CURSOR ROW BASE ADDRESS FROM INDEX "ROW"
81E2:           *
81E2:A6 95      CRADDR  LDX  ROW        ; GET INDEX
81E4:BD 2F 83           LDA  CHLIST,X   ; LOOKUP HIGH PART
81E7:85 8D              STA  CRBASE+1   ; SAVE IN HIGH POINTER
81E9:BD 47 83           LDA  CLLIST,X   ; LOOKUP LOW PART
81EC:85 8C              STA  CRBASE      ; SAVE IN LOW POINTER
81EE:60                 RTS
81EF:           *
81EF:           *
81EF:           * ROUTINE TO SEE IF CURSOR IS IN A NUMERITC FIELD
81EF:           *
81EF:A6 90      NUMFLD  LDX  CROW       ; LOOKUP ROW BASE ADDRESS
81F1:BD DF 8A           LDA  FHLIST,X   ; HIGH PART OF FIELD LIST
81F4:85 8F              STA  FBASE+1    ; SAVE IN HIGH POINTER
81F6:BD F7 8A           LDA  FLLIST,X   ; LOOKUP LOW PART
81F9:85 8E              STA  FBASE      ; SAVE IN LOW POINTER
81FB:A4 91              LDY  CCOL       ; GET THE COLUMN NUMBER
81FD:B1 8E              LDA  (FBASE),Y  ; GET THE FIELD FLAG
81FF:4C 1F 81           JMP  HEXKEY     ; GO SEE IF LEGAL HEX DIGIT
8202:           *
8202:18         NONUM   CLC             ; CLEAR CARRY TO RETURN "FALSE"
8203:60                 RTS
8204:           *
8204:           *
8204:           * ROUTINE TO HANDLE SPECIAL CASE
8204:A5 93      SPECCK  LDA  OLDCR      ; SEE IF IN SPECIAL ROW
8206:C9 12              CMP  #$12
8208:D0 15              BNE  MTEST      ; CHECK FOR MASK UPDATE
820A:A5 94              LDA  OLDCCL     ; ELSE CHECK COLUMN FOR LAST ADDRESS DIGIT
820C:C9 15              CMP  #$15
820E:F0 0D              BEQ  SPDONE     ; IF SO WE HAVE DONE THE ADDRESS
8210:C9 1C              CMP  #$1C       ; THIS IS THE WRITE OPERATION COLUMN
8212:D0 16              BNE  NOSPEC     ; IF NOT THIS THEN NOT SPECIAL
8214:20 74 80           JSR  SPECR      ; DO A SPECIAL READ TO SET UP THE ADDRESS
8217:20 5E 82           JSR  GETVAL     ; GET THE VALUE TO WRITE
821A:20 8D 82           JSR  WTREG      ; DO THE WRITE
821D:38         SPDONE  SEC             ; DONE SO SET CARRY AND RETURN
821E:60                 RTS
821F:           *
```

```
821F:C9 13     MTEST   CMP  #$13      ; SEE IF AT READ MASK ROW
8221:D0 07             BNE  NOSPEC    ; IF NOT THEN NOT SPECIAL
8223:20 5E 82          JSR  GETVAL    ; ELSE GET VALUE OF NEW MASK
8226:85 9B             STA  MASK      ; SAVE IT
8228:38                SEC            ; RETURN MARKING SPECIAL
8229:60                RTS
822A:          *
822A:18        NOSPEC  CLC            ; IF NOT SPECIAL THEN CLEAR CARRY AND RETURN
822B:60                RTS
822C:          *
822C:          * ROUTINE TO PUT CHARACTOR AT CURSOR
822C:          *
822C:A5 90     DSPCHR  LDA  CROW      ; GET THE CURSOR ROW
822E:85 95             STA  ROW       ; SETUP FOR ROW ADDRESS COMPUTATION
8230:20 E2 81          JSR  CRADDR    ; DO LOOKUP
8233:A4 91             LDY  CCOL      ; GET THE COLUMN
8235:A5 99             LDA  LSTKEY    ; GET THE CHARACTOR
8237:91 8C             STA  (CRBASE),Y ; PUT IT ON THE SCREEN
8239:60                RTS
823A:          *
823A:          *
823A:          * ROUTINE TO DO THE VCS REG WRITE
823A:          *
823A:20 4A 82  DOWRT   JSR  GETFLD    ; GET THE FIELD BYTE FOR THE OLD CURSOR
823D:20 57 82          JSR  SETREG    ; TURN THAT INTO A VCS REGISTER ADDRESS
8240:20 CC 80          JSR  SETADR    ; SET IT UP IN THE VCS
8243:20 5E 82          JSR  GETVAL    ; GO GET THE CURRENT VALUE OF THE OLD CURSOR FIELD
8246:20 8D 82          JSR  WTREG     ; GO SEND IT TO THE VCS
8249:60                RTS              ; DONE
824A:          *
824A:          *
824A:          * ROUTINE TO GET FIELD BYTE
824A:          *
824A:A6 93     GETFLD  LDX  OLDCR     ; GET THE OLD CURSOR ROW
824C:BD DF 8A          LDA  FHLIST,X  ; GET THE HIGH PART OF THE FIELD ROW BASE ADDRESS
824F:85 8D             STA  CRBASE+1  ; USE THIS POINTER SO WE CAN DROP IN BELOW
8251:BD F7 8A          LDA  FLLIST,X  ; NOW THE LOW PART
8254:4C 68 82          JMP  GVAL1       ; DO THE REST BELOW
8257:          *
8257:          *
8257:          * ROUTINE TO GO FROM FIELD BYTE TO VCS ADDRESS
8257:          *
8257:85 98     SETREG  STA  RDLOW     ; SAVE IN LOW ORDER REGISTER
8259:A9 00             LDA  #0        ; CLEAR OUT HIGH PART
825B:85 97             STA  RDHGH
825D:60                RTS
825E:          *
825E:          *
825E:          * ROUTINE TO GET VALUE OF THE FIELD AT OLD CURSOR
825E:          *
825E:A6 93     GETVAL  LDX  OLDCR     ; GET THE OLD CURSOR ROW
8260:BD 2F 83          LDA  CHLIST,X  ; GET HIGH PART OF ROW BASE ADDRESS
8263:85 8D             STA  CRBASE+1, ; SAVE IN HIGH PART OF POINTER
8265:BD 47 83          LDA  CLLIST,X  ; GET LOW PART OF ROW BASE ADDRESS
8268:85 8C     GVAL1   STA  CRBASE    ; SAVE IT
826A:A4 94             LDY  OLDCCL    ; GET THE OLD CURSOR COLUMN POSITION
826C:B1 8C             LDA  (CRBASE),Y ; THIS GETS THE LOW DIGIT OF THE VALUE
826E:48                PHA              ; SAVE IT
826F:88                DEY              ; BACK UP TO THE HIGH DIGIT
8270:B1 8C             LDA  (CRBASE),Y ; GET THAT
8272:20 82 82          JSR  TOBIN     ; CONVERT FROM HEX DIGIT TO BINARY
8275:0A                ASL  A         ; MAKE HIGH PART
8276:0A                ASL  A
8277:0A                ASL  A
8278:0A                ASL  A
8279:85 96             STA  TMP1      ; SAVE FOR LATER
827B:68                PLA            ; GET LOW ORDER HEX DIGIT BACK
827C:20 82 82          JSR  TOBIN     ; CONVERT TO BINARY
827F:05 96             ORA  TMP1      ; THIS PUTS IT TOGETHER
8281:60                RTS
8282:          *
8282:          *
8282:          * ROUTINE TO CONVERT ASCII FOR HEX DIGIT BACK TO BINARY
8282:          *
8282:29 7F     TOBIN   AND  #$7F      ; GET RID OF TOP BIT
8284:C9 3A             CMP  #$3A      ; SEE IF IN NUMERIC RANGE
8286:90 02             BCC  TOB1      ; BRANCH IF SO
```

```
8288:E9 07      461         SBC  #7        ; ELSE SUBTRACT 7 TO CORRECT ALPHA RANGE
828A:29 0F      462 TOB1    AND  #$0F      ; NOW WE JUST NEED THE LAST 4 BITS
828C:60         463         RTS
828D:           464 *
828D:           465 *
828D:           466 * ROUTINE TO DO A VCS REGISTER WRITE
828D:           467 *
828D:48         468 WTREG   PHA            ; SAVE VALUE
828E:A0 00      469         LDY  #0        ; SET UP TO INDIRECT
8290:B1 86      470 WTR1    LDA  (FSTAT),Y ; LOOK AT FROB STATUS REGISTER
8292:10 FC      471         BPL  WTR1      ; WAIT TIL OK TO WRITE
8294:A9 40      472         LDA  #WRCMD    ; SEND IT A FROB EXPLORER WRITE COMMAND
8296:91 88      473         STA  (FDATA),Y
8298:B1 86      474 WTR2    LDA  (FSTAT),Y ; WAIT FOR IT TO GET IT
829A:10 FC      475         BPL  WTR2
829C:68         476         PLA            ; GET VALUE BACK
829D:91 88      477         STA  (FDATA),Y SEND IT TO VCS
829F:60         478         RTS
82A0:           479 *
82A0:           480 *
82A0:           481 * THE INITILAIZATION ROUTINE
82A0:           482 *
82A0:20 E7 82   483 INIT    JSR  SLOAD     ; PUT UP THE SCREEN
82A3:A9 C0      484         LDA  #$C0      ; SET UP FSTAT AND FDATA
82A5:85 87      485         STA  FSTAT+1
82A7:85 89      486         STA  FDATA+1
82A9:A9 A0      487 PARAM1  LDA  #$A0
82AB:85 86      488         STA  FSTAT
82AD:A9 A1      489 PARAM2  LDA  #$A1
82AF:85 88      490         STA  FDATA
82B1:A9 00      491         LDA  #0        ; CLEAR SOME VARIABLES
82B3:85 94      492         STA  OLDCCL
82B5:85 93      493         STA  OLDCR
82B7:85 91      494         STA  CCOL
82B9:85 90      495         STA  CROW
82BB:85 99      496         STA  LSTKEY
82BD:A9 81      497         LDA  #$81      ; INIT THE MASK
82BF:85 9B      498         STA  MASK
82C1:           499 *
82C1:           500 * NOW A LOOP TO DO THE INITIAL REGISTER WRITES
82C1:           501 *
82C1:20 EF 81   502 ILP     JSR  NUMFLD    ; IS CURSOR AT A NUMBER FIELD?
82C4:B0 1A      503              BCS       INFLD ; IF YES, JUST PUT THE FLAG IN LSTKEY AND MOVE ON
82C6:24 99      504         BIT  LSTKEY    ; IF NOT, SEE IF WE JUST WERE
82C8:F0 07      505         BEQ  NOCHNG    ; IF NOT THEN THE CHANGE IS NOT IMPORTANT
82CA:A9 00      506         LDA  #0        ; CLEAR LSTKEY FLAG
82CC:85 99      507         STA  LSTKEY
82CE:20 3A 82   508         JSR  DOWRT     ; UPDATE THE REGISTER
82D1:           509 *
82D1:20 90 81   510 NOCHNG  JSR  MOVRGH    ; MOVE TO THE NEXT POSITION
82D4:A5 91      511         LDA  CCOL      ; GET NEW CURSOR COLUMN
82D6:D0 E9      512         BNE  ILP       ; LOOP IF NOT READY FOR NEW LINE
82D8:20 7C 81   513         JSR  MOVDWN    ; ELSE MOVE DOWN TO NEXT LINE
82DB:A5 90      514         LDA  CROW      ; SEE IF AT THE END OF THE DISPLAY
82DD:D0 E2      515         BNE  ILP       ; IF NOT KEEP GOING
82DF:60         516         RTS              ; BUT IF SO THEN WE ARE DONE
82E0:           517 *
82E0:A9 FF      518 INFLD   LDA  #$FF      ; SET LSTKEY TO INDICATE WE ARE IN A FIELD
82E2:85 99      519         STA  LSTKEY
82E4:4C D1 82   520         JMP  NOCHNG    ; GET BACK IN
82E7:           521 *
82E7:           522 *
82E7:           523 *
82E7:           524 * ROUTINE TO LOAD DISPLAY FROM ARRAY "SCREEN"
82E7:           525 *
82E7:A2 17      526 SLOAD   LDX  #$17      ; DO FOR 24 LINES
82E9:A9 86      527         LDA  #<SCREEN+920 ; SCREEN+(23*40) POINTS TO THE LAST LINE
82EB:85 8B      528         STA  RBASE+1   ; STORE THE HIGH PART
82ED:A9 F7      529         LDA  #>SCREEN+920 ; NOW THE LOW PART
82EF:85 8A      530         STA  RBASE
82F1:BD 2F 83   531 SL1     LDA  CHLIST,X  ; GET HIGH POINTER TO DISPLAY ROW
82F4:85 8D      532         STA  CRBASE+1  ; PUT IN POINTER
82F6:BD 47 83   533         LDA  CLLIST,X  ; GET LOW PART
82F9:85 8C      534         STA  CRBASE    ; PUT THAT IN POINTER
82FB:A0 27      535         LDY  #39       ; WORK BACK FROM END OF LINE
82FD:B1 8A      536 SL2     LDA  (RBASE),Y ; GET CHARACTOR FROM ARRAY
82FF:91 8C      537         STA  (CRBASE),Y ; PUT UP ON SCREEN
```

```
8301:88                DEY
8302:10 F9             BPL  SL2       ; DO FOR Y=39 TO 0
8304:38                SEC            ; SUBTRACT 40 FROM RBASE
8305:A5 8A             LDA  RBASE
8307:E9 28             SBC  #40
8309:85 8A             STA  RBASE
830B:A5 8B             LDA  RBASE+1   ; 16 BIT
830D:E9 00             SBC  #0
830F:85 8B             STA  RBASE+1
8311:CA                DEX              ; THIS IS THE MAJOR LOOP COUNT
8312:10 DD             BPL  SL1       ; WORK THROUGH 24 LINES
8314:60                RTS
8315:          *
8315:          *
8315:          SAVER   EQU  *         ; ROUTINE TO SAVE PAGE 0 LOCATIONS FOR BASIC
8315:A2 80             LDX  #$80      ; START AT LOC 80 HEX
8317:B5 00     SAVLP   LDA  0,X       ; GET BYTE
8319:9D 8F 8A          STA  SAVBUF-$80,X ; SAVE IT
831C:E8                INX            ; MOVE TO NEXT
831D:30 F8             BMI  SAVLP     ; KEEP GOING
831F:4C A0 82          JMP  INIT      ; THE REST START IN INIT
8322:          *
8322:          *
8322:          RESTORER EQU *         ; ROUTINE TO PUT BACK THE BASIC LOCATIONS TO PAGE 0
8322:A2 80             LDX  #$80      ; START AT LOC 80 HEX
8324:BD 8F 8A  RESLP   LDA  SAVBUF-$80,X ; GET BYTE
8327:95 00             STA  0,X       ; RESTORE IT
8329:E8                INX            ; MOVE TO NEXT
832A:30 F8             BMI  RESLP     ; DO FOR X=$80 TO $FF
832C:4C D0 03          JMP  $3D0      ; RETURN TO BASIC
832F:          *
832F:          *
832F:          *
832F:          * SCREEN ROW BASE LOOKUP TABLES
832F:          *
832F:04 04 05  CHLIST  DFB  4,4,5,5,6,6,7,7,4,4,5,5,6,6,7,7,4,4,5,5,6,6,7,7
8332:05 06 06
8335:07 07 04
8338:04 05 05
833B:06 06 07
833E:07 04 04
8341:05 05 06
8344:06 07 07
8347:00 80 00  CLLIST  DFB  0,$80,0,$80,0,$80,0,$80
834A:80 00 80
834D:00 80
834F:28 A8 28          DFB  $28,$A8,$28,$A8,$28,$A8,$28,$A8
8352:A8 28 A8
8355:28 A8
8357:50 D0 50          DFB  $50,$D0,$50,$D0,$50,$D0,$50,$D0
835A:D0 50 D0
835D:50 D0
835F:          *
835F:          *
835F:          * HERE IS THE PAGE SETUP
835F:          *
835F:                  MSB  ON        ; DO STRINGS WITH MSB ON
835F:          *
835F:A0 A0 A0  SCREEN  ASC  /              FROB EXPLORER CONTROL SCREEN      /
8362:A0 A0 A0
8365:C6 D2 CF
8368:C2 A0 C5
836B:D8 D0 CC
836E:CF D2 C5
8371:D2 A0 C3
8374:CF CE D4
8377:D2 CF CC
837A:A0 D3 C3
837D:D2 C5 C5
8380:CE A0 A0
8383:A0 A0 A0
8386:A0
8387:C5 CE C1          ASC  /ENABLE:0C  (PUT 40 HERE FOR SCREEN2)    /
838A:C2 CC C5
838D:BA B0 C3
8390:A0 A0 A8
8393:D0 D5 D4
```

```
8396:A0 B4 B0
8399:A0 C8 C5
839C:D2 C5 A0
839F:C6 CF D2
83A2:A0 D3 C3
83A5:D2 C5 C5
83A8:CE B2 A9
83AB:A0 A0 A0
83AE:A0
83AF:D0 CC C1  585         ASC  /PLAYER1:COLOR 82 VERT 14 HORZ 30 FLP 00 /
83B2:D9 C5 D2
83B5:B1 BA C3
83B8:CF CC CF
83BB:D2 A0 B8
83BE:B2 A0 D6
83C1:C5 D2 D4
83C4:A0 B1 B4
83C7:A0 C8 CF
83CA:D2 DA A0
83CD:B3 B0 A0
83D0:C6 CC D0
83D3:A0 B0 B0
83D6:A0
83D7:A0 A0 A0  586         ASC  /            IMAGE: FE 7F 63 62 7C 78 60 F2      /
83DA:A0 C9 CD
83DD:C1 C7 C5
83E0:BA A0 C6
83E3:C5 A0 B7
83E6:C6 A0 B6
83E9:B3 A0 B6
83EC:B2 A0 B7
83EF:C3 A0 B7
83F2:B8 A0 B6
83F5:B0 A0 C6
83F8:B2 A0 A0
83FB:A0 A0 A0
83FE:A0
83FF:A0 A0 A0  587         ASC  /            SOUND: A:00 B:00 C:00               /
8402:A0 D3 CF
8405:D5 CE C4
8408:BA A0 C1
840B:BA B0 B0
840E:A0 C2 BA
8411:B0 B0 A0
8414:C3 BA B0
8417:B0 A0 A0
841A:A0 A0 A0
841D:A0 A0 A0
8420:A0 A0 A0
8423:A0 A0 A0
8426:A0
8427:D0 CC C1  588         ASC  /PLAYER2:COLOR 00 VERT 1E HORZ 30 FLP 00 /
842A:D9 C5 D2
842D:B2 BA C3
8430:CF CC CF
8433:D2 A0 B0
8436:B0 A0 D6
8439:C5 D2 D4
843C:A0 B1 C5
843F:A0 C8 CF
8442:D2 DA A0
8445:B3 B0 A0
8448:C6 CC D0
844B:A0 B0 B0
844E:A0
844F:A0 A0 A0  589         ASC  /            IMAGE: FE 7F 63 62 7C 78 60 F6      /
8452:A0 C9 CD
8455:C1 C7 C5
8458:BA A0 C6
845B:C5 A0 B7
845E:C6 A0 B6
8461:B3 A0 B6
8464:B2 A0 B7
8467:C3 A0 B7
846A:B8 A0 B6
846D:B0 A0 C6
8470:B6 A0 A0
```

```
8473:A0 A0 A0
8476:A0
8477:A0 A0 A0  590         ASC  /            SOUND: A:00 B:00 C:00              /
847A:A0 D3 CF
847D:D5 CE C4
8480:BA A0 C1
8483:BA B0 B0
8486:A0 C2 BA
8489:B0 B0 A0
848C:C3 BA B0
848F:B0 A0 A0
8492:A0 A0 A0
8495:A0 A0 A0
8498:A0 A0 A0
849B:A0 A0 A0
849E:A0
849F:C2 C1 C3  591         ASC  /BACKGROUND COLOR EA  OBJECT COLOR 64    /
84A2:CB C7 D2
84A5:CF D5 CE
84A8:C4 A0 C3
84AB:CF CC CF
84AE:D2 A0 C5
84B1:C1 A0 A0
84B4:CF C2 CA
84B7:C5 C3 D4
84BA:A0 C3 CF
84BD:CC CF D2
84C0:A0 B6 B4
84C3:A0 A0 A0
84C6:A0
84C7:CF C2 CA  592         ASC  /OBJECT   A IMAGE: 18 24 42 42 7E 42 42 CE /
84CA:C5 C3 D4
84CD:A0 C1 A0
84D0:C9 CD C1
84D3:C7 C5 BA
84D6:A0 B1 B8
84D9:A0 B2 B4
84DC:A0 B4 B2
84DF:A0 B4 B2
84E2:A0 B7 C5
84E5:A0 B4 B2
84E8:A0 B4 B2
84EB:A0 C3 C5
84EE:A0
84EF:CF C2 CA  593         ASC  /OBJECT   B IMAGE: FE 61 61 7E 7E 61 61 FE /
84F2:C5 C3 D4
84F5:A0 C2 A0
84F8:C9 CD C1
84FB:C7 C5 BA
84FE:A0 C6 C5
8501:A0 B6 B1
8504:A0 B6 B1
8507:A0 B7 C5
850A:A0 B7 C5
850D:A0 B6 B1
8510:A0 B6 B1
8513:A0 C6 C5
8516:A0
8517:CF C2 CA  594         ASC  /OBJECT   C IMAGE: 7E FF FE F0 F0 FE FF 7E /
851A:C5 C3 D4
851D:A0 C3 A0
8520:C9 CD C1
8523:C7 C5 BA
8526:A0 B7 C5
8529:A0 C6 C6
852C:A0 C6 C5
852F:A0 C6 B0
8532:A0 C6 B0
8535:A0 C6 C5
8538:A0 C6 C6
853B:A0 B7 C5
853E:A0
853F:D2 C5 C7  595         ASC  /REG03:00 P1VMODE:00 P2VMODE:00 FVMODE:05/
8542:B0 B3 BA
8545:B0 B0 A0
8548:D0 B1 D6
854B:CD CF C4
```

```
854E:C5 BA B0
8551:B0 A0 D0
8554:B2 D6 CD
8557:CF C4 C5
855A:BA B0 B0
855D:A0 C6 D6
8560:CD CF C4
8563:C5 BA B0
8566:B5
8567:D0 B1 C8  596         ASC  /P1HRES:00 P2HRES:00 S1HRES:00 S2HRES:00 /
856A:D2 C5 D3
856D:BA B0 B0
8570:A0 D0 B2
8573:C8 D2 C5
8576:D3 BA B0
8579:B0 A0 D3
857C:B1 C8 D2
857F:C5 D3 BA
8582:B0 B0 A0
8585:D3 B2 C8
8588:D2 C5 D3
858B:BA B0 B0
858E:A0
858F:D3 B3 C8  597         ASC  /S3HRES:00 S3SHOT:00 S3HORZ:00 P1DELAY:00/
8592:D2 C5 D3
8595:BA B0 B0
8598:A0 D3 B3
859B:D3 C8 CF
859E:D4 BA B0
85A1:B0 A0 D3
85A4:B3 C8 CF
85A7:D2 DA BA
85AA:B0 B0 A0
85AD:D0 B1 C4
85B0:C5 CC C1
85B3:D9 BA B0
85B6:B0
85B7:D0 B2 C4  598         ASC  /P2DELAY:00 REG27:00 S1CONT:00 S2CONT:00 /
85BA:C5 CC C1
85BD:D9 BA B0
85C0:B0 A0 D2
85C3:C5 C7 B2
85C6:B7 BA B0
85C9:B0 A0 D3
85CC:B1 C3 CF
85CF:CE D4 BA
85D2:B0 B0 A0
85D5:D3 B2 C3
85D8:CF CE D4
85DB:BA B0 B0
85DE:A0
85DF:C8 DA D3  599         ASC  /HZSCRL:00 NOINC:00 COLRES:00 REG2D:00   /
85E2:C3 D2 CC
85E5:BA B0 B0
85E8:A0 CE CF
85EB:C9 CE C3
85EE:BA B0 B0
85F1:A0 C3 CF
85F4:CC D2 C5
85F7:D3 BA B0
85FA:B0 A0 D2
85FD:C5 C7 B2
8600:C4 BA B0
8603:B0 A0 A0
8606:A0
8607:D2 C5 C7  600         ASC  /REG2E:00 REG2F:00                       /
860A:B2 C5 BA
860D:B0 B0 A0
8610:D2 C5 C7
8613:B2 C6 BA
8616:B0 B0 A0
8619:A0 A0 A0
861C:A0 A0 A0
861F:A0 A0 A0
8622:A0 A0 A0
8625:A0 A0 A0
8628:A0 A0 A0
```

```
862B:A0 A0 A0
862E:A0
862F:C7 C5 CE  601         ASC  /GENERAL  WRITE LOC:0081 VAL:00  READ=00  /
8632:C5 D2 C1
8635:CC A0 D7
8638:D2 C9 D4
863B:C5 A0 CC
863E:CF C3 BA
8641:B0 B0 B8
8644:B1 A0 D6
8647:C1 CC BA
864A:B0 B0 A0
864D:A0 D2 C5
8650:C1 C4 BD
8653:B0 B0 A0
8656:A0
8657:D2 C5 C1  602         ASC  /READ     ONLY LOCATIONS WITH MASK:81        /
865A:C4 A0 CF
865D:CE CC D9
8660:A0 CC CF
8663:C3 C1 D4
8666:C9 CF CE
8669:D3 A0 D7
866C:C9 D4 C8
866F:A0 CD C1
8672:D3 CB BA
8675:B8 B1 A0
8678:A0 A0 A0
867B:A0 A0 A0
867E:A0
867F:D2 C5 C7  603         ASC  /REG30:00 REG31:00 REG32:00 REG33:00     /
8682:B3 B0 BA
8685:B0 B0 A0
8688:D2 C5 C7
868B:B3 B1 BA
868E:B0 B0 A0
8691:D2 C5 C7
8694:B3 B2 BA
8697:B0 B0 A0
869A:D2 C5 C7
869D:B3 B3 BA
86A0:B0 B0 A0
86A3:A0 A0 A0
86A6:A0
86A7:D2 C5 C7  604         ASC  /REG34:00 REG35:00 REG36:00 REG37:00     /
86AA:B3 B4 BA
86AD:B0 B0 A0
86B0:D2 C5 C7
86B3:B3 B5 BA
86B6:B0 B0 A0
86B9:D2 C5 C7
86BC:B3 B6 BA
86BF:B0 B0 A0
86C2:D2 C5 C7
86C5:B3 B7 BA
86C8:B0 B0 A0
86CB:A0 A0 A0
86CE:A0
86CF:D2 C5 C7  605         ASC  /REG38:00 REG39:00 REG3A:00 REG3B:00     /
86D2:B3 B8 BA
86D5:B0 B0 A0
86D8:D2 C5 C7
86DB:B3 B9 BA
86DE:B0 B0 A0
86E1:D2 C5 C7
86E4:B3 C1 BA
86E7:B0 B0 A0
86EA:D2 C5 C7
86ED:B3 C2 BA
86F0:B0 B0 A0
86F3:A0 A0 A0
86F6:A0
86F7:D2 C5 C7  606         ASC  /REG3C:00 REG3D:00 REG3E:00 REG3F:00     /
86FA:B3 C3 BA
86FD:B0 B0 A0
8700:D2 C5 C7
8703:B3 C4 BA
```

```
8706:B0 B0 A0
8709:D2 C5 C7
870C:B3 C5 BA
870F:B0 B0 A0
8712:D2 C5 C7
8715:B3 C6 BA
8718:B0 B0 A0
871B:A0 A0 A0
871E:A0
871F:          607 *
871F:          608 *
871F:A0 A0 A0  609 SR0     ASC  /              XXXX XXXXXXXX XXXXXXX XXXXXX      /
8722:A0 A0 A0
8725:D8 D8 D8
8728:D8 A0 D8
872B:D8 D8 D8
872E:D8 D8 D8
8731:D8 A0 D8
8734:D8 D8 D8
8737:D8 D8 D8
873A:A0 D8 D8
873D:D8 D8 D8
8740:D8 A0 A0
8743:A0 A0 A0
8746:A0
8747:D8 D8 D8  610 SR1     ASC  /XXXXXX:AA                              /
874A:D8 D8 D8
874D:BA C1 C1
8750:A0 A0 A0
8753:A0 A0 A0
8756:A0 A0 A0
8759:A0 A0 A0
875C:A0 A0 A0
875F:A0 A0 A0
8762:A0 A0 A0
8765:A0 A0 A0
8768:A0 A0 A0
876B:A0 A0 A0
876E:A0
876F:D8 D8 D8  611 SR2     ASC  /XXXXXXX:XXXXX AB XXXX AF XXXX 89 XXX 0B /
8772:D8 D8 D8
8775:D8 BA D8
8778:D8 D8 D8
877B:D8 A0 C1
877E:C2 A0 D8
8781:D8 D8 D8
8784:A0 C1 C6
8787:A0 D8 D8
878A:D8 D8 A0
878D:B8 B9 A0
8790:D8 D8 D8
8793:A0 B0 C2
8796:A0
8797:A0 A0 A0  612 SR3     ASC  /            XXXXX: B3 B4 B5 B6 B7 B8 B9 BA      /
879A:A0 D8 D8
879D:D8 D8 D8
87A0:BA A0 C2
87A3:B3 A0 C2
87A6:B4 A0 C2
87A9:B5 A0 C2
87AC:B6 A0 C2
87AF:B7 A0 C2
87B2:B8 A0 C2
87B5:B9 A0 C2
87B8:C1 A0 A0
87BB:A0 A0 A0
87BE:A0
87BF:A0 A0 A0  613 SR4     ASC  /            XXXXX: X:15 X:17 X:19               /
87C2:A0 D8 D8
87C5:D8 D8 D8
87C8:BA A0 D8
87CB:BA B1 B5
87CE:A0 D8 BA
87D1:B1 B7 A0
87D4:D8 BA B1
87D7:B9 A0 A0
87DA:A0 A0 A0
```

```
87DD:A0 A0 A0
87E0:A0 A0 A0
87E3:A0 A0 A0
87E6:A0
87E7:D8 D8 D8  614 SR5     ASC  /XXXXXXX:XXXXX AC XXXX B0 XXXX 8A XXX 0C /
87EA:D8 D8 D8
87ED:D8 BA D8
87F0:D8 D8 D8
87F3:D8 A0 C1
87F6:C3 A0 D8
87F9:D8 D8 D8
87FC:A0 C2 B0
87FF:A0 D8 D8
8802:D8 D8 A0
8805:B8 C1 A0
8808:D8 D8 D8
880B:A0 B0 C3
880E:A0
880F:A0 A0 A0  615 SR6     ASC  /            XXXXX: BB BC BD BE BF C0 C1 C2      /
8812:A0 D8 D8
8815:D8 D8 D8
8818:BA A0 C2
881B:C2 A0 C2
881E:C3 A0 C2
8821:C4 A0 C2
8824:C5 A0 C2
8827:C6 A0 C3
882A:B0 A0 C3
882D:B1 A0 C3
8830:B2 A0 A0
8833:A0 A0 A0
8836:A0
8837:A0 A0 A0  616 SR7     ASC  /            XXXXX: X:16 X:18 X:1A               /
883A:A0 D8 D8
883D:D8 D8 D8
8840:BA A0 D8
8843:BA B1 B6
8846:A0 D8 BA
8849:B1 B8 A0
884C:D8 BA B1
884F:C1 A0 A0
8852:A0 A0 A0
8855:A0 A0 A0
8858:A0 A0 A0
885B:A0 A0 A0
885E:A0
885F:D8 D8 D8  617 SR8     ASC  /XXXXXXXXXX XXXXX AD  XXXXXX XXXXX AE    /
8862:D8 D8 D8
8865:D8 D8 D8
8868:D8 A0 D8
886B:D8 D8 D8
886E:D8 A0 C1
8871:C4 A0 A0
8874:D8 D8 D8
8877:D8 D8 D8
887A:A0 D8 D8
887D:D8 D8 D8
8880:A0 C1 C5
8883:A0 A0 A0
8886:A0
8887:D8 D8 D8  618 SR9     ASC  /XXXXXX   X XXXXX: C4 C5 C6 C7 C8 C9 CA CB /
888A:D8 D8 D8
888D:A0 D8 A0
8890:D8 D8 D8
8893:D8 D8 BA
8896:A0 C3 B4
8899:A0 C3 B5
889C:A0 C3 B6
889F:A0 C3 B7
88A2:A0 C3 B8
88A5:A0 C3 B9
88A8:A0 C3 C1
88AB:A0 C3 C2
88AE:A0
88AF:D8 D8 D8  619 SRA     ASC  /XXXXXX   X XXXXX: CD CE CF D0 D1 D2 D3 D4 /
88B2:D8 D8 D8
88B5:A0 D8 A0
```

```
88B8:D8 D8 D8
88BB:D8 D8 BA
88BE:A0 C3 C4
88C1:A0 C3 C5
88C4:A0 C3 C6
88C7:A0 C4 B0
88CA:A0 C4 B1
88CD:A0 C4 B2
88D0:A0 C4 B3
88D3:A0 C4 B4
88D6:A0
88D7:D8 D8 D8  620 SRB     ASC  /XXXXXX   X XXXXX: D6 D7 D8 D9 DA DB DC DD /
88DA:D8 D8 D8
88DD:A0 D8 A0
88E0:D8 D8 D8
88E3:D8 D8 BA
88E6:A0 C4 B6
88E9:A0 C4 B7
88EC:A0 C4 B8
88EF:A0 C4 B9
88F2:A0 C4 C1
88F5:A0 C4 C2
88F8:A0 C4 C3
88FB:A0 C4 C4
88FE:A0
88FF:D8 D8 D8  621 SRC     ASC  /XXXXX:03 XXXXXXX:04 XXXXXXX:05 XXXXXX:A5/
8902:D8 D8 BA
8905:B0 B3 A0
8908:D8 D8 D8
890B:D8 D8 D8
890E:D8 BA B0
8911:B4 A0 D8
8914:D8 D8 D8
8917:D8 D8 D8
891A:BA B0 B5
891D:A0 D8 D8
8920:D8 D8 D8
8923:D8 BA C1
8926:B5
8927:D8 D8 D8  622 SRD     ASC  /XXXXXX:10 XXXXX:11 XXXXXX:12 XXXXXX:13 /
892A:D8 D8 D8
892D:BA B1 B0
8930:A0 D8 D8
8933:D8 D8 D8
8936:D8 BA B1
8939:B1 A0 D8
893C:D8 D8 D8
893F:D8 D8 BA
8942:B1 B2 A0
8945:D8 D8 D8
8948:D8 D8 D8
894B:BA B1 B3
894E:A0
894F:D8 D8 D8  623 SRE     ASC  /XXXXXX:14 XXXXXX:1F XXXXXX:24 XXXXXXX:25/
8952:D8 D8 D8
8955:BA B1 B4
8958:A0 D8 D8
895B:D8 D8 D8
895E:D8 BA B1
8961:C6 A0 D8
8964:D8 D8 D8
8967:D8 D8 BA
896A:B2 B4 A0
896D:D8 D8 D8
8970:D8 D8 D8
8973:D8 BA B2
8976:B5
8977:D8 D8 D8  624 SRF     ASC  /XXXXXXX:26 XXXXX:27 XXXXXX:28 XXXXXX:29 /
897A:D8 D8 D8
897D:D8 BA B2
8980:B6 A0 D8
8983:D8 D8 D8
8986:D8 BA B2
8989:B7 A0 D8
898C:D8 D8 D8
898F:D8 D8 BA
8992:B2 B8 A0
```

```
8995:D8 D8 D8
8998:D8 D8 D8
899B:BA B2 B9
899E:A0
899F:D8 D8 D8  625 SR10    ASC  /XXXXXX:2A XXXXX:2B XXXXXX:2C XXXXX:2D   /
89A2:D8 D8 D8
89A5:BA B2 C1
89A8:A0 D8 D8
89AB:D8 D8 D8
89AE:BA B2 C2
89B1:A0 D8 D8
89B4:D8 D8 D8
89B7:D8 BA B2
89BA:C3 A0 D8
89BD:D8 D8 D8
89C0:D8 BA B2
89C3:C4 A0 A0
89C6:A0
89C7:D8 D8 D8  626 SR11    ASC  /XXXXX:2E XXXXX:2F                       /
89CA:D8 D8 BA
89CD:B2 C5 A0
89D0:D8 D8 D8
89D3:D8 D8 BA
89D6:B2 C6 A0
89D9:A0 A0 A0
89DC:A0 A0 A0
89DF:A0 A0 A0
89E2:A0 A0 A0
89E5:A0 A0 A0
89E8:A0 A0 A0
89EB:A0 A0 A0
89EE:A0
89EF:D8 D8 D8  627 SR12    ASC  /XXXXXXX  XXXXX XXX:0000 XXX:00  XXXX=XX  /
89F2:D8 D8 D8
89F5:D8 A0 D8
89F8:D8 D8 D8
89FB:D8 A0 D8
89FE:D8 D8 BA
8A01:B0 B0 B0
8A04:B0 A0 D8
8A07:D8 D8 BA
8A0A:B0 B0 A0
8A0D:A0 D8 D8
8A10:D8 D8 BD
8A13:D8 D8 A0
8A16:A0
8A17:D8 D8 D8  628 SR13    ASC  /XXXX     XXXX XXXXXXXXX XXXX XXXX:00        /
8A1A:D8 A0 D8
8A1D:D8 D8 D8
8A20:A0 D8 D8
8A23:D8 D8 D8
8A26:D8 D8 D8
8A29:D8 A0 D8
8A2C:D8 D8 D8
8A2F:A0 D8 D8
8A32:D8 D8 BA
8A35:B0 B0 A0
8A38:A0 A0 A0
8A3B:A0 A0 A0
8A3E:A0
8A3F:D8 D8 D8  629 SR14    ASC  /XXXXX:30 XXXXX:31 XXXXX:32 XXXXX:33     /
8A42:D8 D8 BA
8A45:B3 B0 A0
8A48:D8 D8 D8
8A4B:D8 D8 BA
8A4E:B3 B1 A0
8A51:D8 D8 D8
8A54:D8 D8 BA
8A57:B3 B2 A0
8A5A:D8 D8 D8
8A5D:D8 D8 BA
8A60:B3 B3 A0
8A63:A0 A0 A0
8A66:A0
8A67:D8 D8 D8  630 SR15    ASC  /XXXXX:34 XXXXX:35 XXXXX:36 XXXXX:37     /
8A6A:D8 D8 BA
8A6D:B3 B4 A0
```

```
8A70:D8 D8 D8
8A73:D8 D8 BA
8A76:B3 B5 A0
8A79:D8 D8 D8
8A7C:D8 D8 BA
8A7F:B3 B6 A0
8A82:D8 D8 D8
8A85:D8 D8 BA
8A88:B3 B7 A0
8A8B:A0 A0 A0
8A8E:A0
8A8F:D8 D8 D8  631 SR16    ASC  /XXXXX:38 XXXXX:39 XXXXX:3A XXXXX:3B     /
8A92:D8 D8 BA
8A95:B3 B8 A0
8A98:D8 D8 D8
8A9B:D8 D8 BA
8A9E:B3 B9 A0
8AA1:D8 D8 D8
8AA4:D8 D8 BA
8AA7:B3 C1 A0
8AAA:D8 D8 D8
8AAD:D8 D8 BA
8AB0:B3 C2 A0
8AB3:A0 A0 A0
8AB6:A0
8AB7:D8 D8 D8  632 SR17    ASC  /XXXXX:3C XXXXX:3D XXXXX:3E XXXXX:3F     /
8ABA:D8 D8 BA
8ABD:B3 C3 A0
8AC0:D8 D8 D8
8AC3:D8 D8 BA
8AC6:B3 C4 A0
8AC9:D8 D8 D8
8ACC:D8 D8 BA
8ACF:B3 C5 A0
8AD2:D8 D8 D8
8AD5:D8 D8 BA
8AD8:B3 C6 A0
8ADB:A0 A0 A0
8ADE:A0
8ADF:          633 *
8ADF:          634 *
8ADF:87 87 87  635 FHLIST  DFB  <SR0,<SR1,<SR2,<SR3,<SR4,<SR5
8AE2:87 87 87
8AE5:88 88 88  636         DFB  <SR6,<SR7,<SR8,<SR9,<SRA,<SRB
8AE8:88 88 88
8AEB:88 89 89  637         DFB  <SRC,<SRD,<SRE,<SRF,<SR10,<SR11
8AEE:89 89 89
8AF1:89 8A 8A  638         DFB  <SR12,<SR13,<SR14,<SR15,<SR16,<SR17
8AF4:8A 8A 8A
8AF7:          639 *
8AF7:1F 47 6F  640 FLLIST  DFB  >SR0,>SR1,>SR2,>SR3,>SR4,>SR5
8AFA:97 BF E7
8AFD:0F 37 5F  641         DFB  >SR6,>SR7,>SR8,>SR9,>SRA,>SRB
8B00:87 AF D7
8B03:FF 27 4F  642         DFB  >SRC,>SRD,>SRE,>SRF,>SR10,>SR11
8B06:77 9F C7
8B09:EF 17 3F  643         DFB  >SR12,>SR13,>SR14,>SR15,>SR16,>SR17
8B0C:67 8F B7
8B0F:          644 *
8B0F:          645 *
8B0F:          646 SAVBUF  DS   $80       ; SPACE FOR PAGE 0 STUFF WE NEED TO SAVE AND RESTORE
8B8F:          647 *
8B8F:          648 *
8B8F:          649 * END OF XCONRTOL

\007*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
 80FB ADRCNV           91 CCOL           8158 CHKKEY         832F CHLIST
 815F CKY1           8347 CLLIST         81E2 CRADDR           8C CRBASE
   90 CROW           81BA CUPDTE         823A DOWRT          8134 DSPBYT
 822C DSPCHR           8E FBASE            88 FDATA          8ADF FHLIST
 8AF7 FLLIST           86 FSTAT          824A GETFLD         825E GETVAL
 8268 GVAL1          811F HEXKEY         8128 HK1            82C1 ILP
 80C3 INCR1          80B7 INCRR          82E0 INFLD          82A0 INIT
 C000 KBD            C010 KSTRB            99 LSTKEY           9A LSTRD
   9B MASK           8189 MDN1           81B5 MLF1           8003 MLOOP
 800B MLP1           8026 MLP2           817C MOVDWN         81A8 MOVLFT
 8190 MOVRGH         8167 MOVUP          819F MRT0           81A1 MRT1
 821F MTEST          8175 MUP1           82D1 NOCHNG         8132 NOHEX
?8202 NONUM          822A NOSPEC         803C NTDWN          8046 NTLFT
 8050 NTRGH          8032 NTUP           81EF NUMFLD           94 OLDCCL
   93 OLDCR         ?82A9 PARAM1        ?82AD PARAM2           8A RBASE
   20 RDCMD            97 RDHGH            98 RDLOW          80EA RDR0
 80F2 RDR1           80E7 RDREG          8324 RESLP          8322 RESTORER
   85 RLOC             95 ROW              92 RRCOL          80CE SA1
 80D6 SA2            80DE SA3              10 SACMD          8B0F SAVBUF
 8315 SAVER          8317 SAVLP          835F SCREEN         80CC SETADR
 8257 SETREG         82F1 SL1            82FD SL2            82E7 SLOAD
 821D SPDONE         8204 SPECCK         8074 SPECR          871F SR0
 8747 SR1            899F SR10           89C7 SR11           89EF SR12
 8A17 SR13           8A3F SR14           8A67 SR15           8A8F SR16
 8AB7 SR17           876F SR2            8797 SR3            87BF SR4
 87E7 SR5            880F SR6            8837 SR7            885F SR8
 8887 SR9            88AF SRA            88D7 SRB            88FF SRC
 8927 SRD            894F SRE            8977 SRF           ?8000 START
 8154 TH1              96 TMP1           828A TOB1           8282 TOBIN
 814C TOHEX            40 WRCMD          8290 WTR1           8298 WTR2
 828D WTREG
```

```
   10 SACMD            20 RDCMD            40 WRCMD            85 RLOC
   86 FSTAT            88 FDATA            8A RBASE            8C CRBASE
   8E FBASE            90 CROW             91 CCOL             92 RRCOL
   93 OLDCR            94 OLDCCL           95 ROW              96 TMP1
   97 RDHGH            98 RDLOW            99 LSTKEY           9A LSTRD
   9B MASK          ?8000 START          8003 MLOOP          800B MLP1
 8026 MLP2           8032 NTUP           803C NTDWN          8046 NTLFT
 8050 NTRGH          8074 SPECR          80B7 INCRR          80C3 INCR1
 80CC SETADR         80CE SA1            80D6 SA2            80DE SA3
 80E7 RDREG          80EA RDR0           80F2 RDR1           80FB ADRCNV
 811F HEXKEY         8128 HK1            8132 NOHEX          8134 DSPBYT
 814C TOHEX          8154 TH1            8158 CHKKEY         815F CKY1
 8167 MOVUP          8175 MUP1           817C MOVDWN         8189 MDN1
 8190 MOVRGH         819F MRT0           81A1 MRT1           81A8 MOVLFT
 81B5 MLF1           81BA CUPDTE         81E2 CRADDR         81EF NUMFLD
?8202 NONUM          8204 SPECCK         821D SPDONE         821F MTEST
 822A NOSPEC         822C DSPCHR         823A DOWRT          824A GETFLD
 8257 SETREG         825E GETVAL         8268 GVAL1          8282 TOBIN
 828A TOB1           828D WTREG          8290 WTR1           8298 WTR2
 82A0 INIT          ?82A9 PARAM1        ?82AD PARAM2         82C1 ILP
 82D1 NOCHNG         82E0 INFLD          82E7 SLOAD          82F1 SL1
 82FD SL2            8315 SAVER          8317 SAVLP          8322 RESTORER
 8324 RESLP          832F CHLIST         8347 CLLIST         835F SCREEN
 871F SR0            8747 SR1            876F SR2            8797 SR3
 87BF SR4            87E7 SR5            880F SR6            8837 SR7
 885F SR8            8887 SR9            88AF SRA            88D7 SRB
 88FF SRC            8927 SRD            894F SRE            8977 SRF
 899F SR10           89C7 SR11           89EF SR12           8A17 SR13
 8A3F SR14           8A67 SR15           8A8F SR16           8AB7 SR17
 8ADF FHLIST         8AF7 FLLIST         8B0F SAVBUF         C000 KBD
 C010 KSTRB
```